TRANSFER AND EINSTELLUNG EFFECTS OF EXAMPLES
ON DEVISING COMPUTER ALGORITHMS

by
WILLIAM PHILIP GODDARD
B. Ed., University of British Columbia, 1963

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ARTS

in the Department
of
Mathematics Education

We accept this thesis as conforming to the
required standard

THE UNIVERSITY OF BRITISH COLUMBIA
SEPTEMBER, 1976

© William Philip Goddard, 1976
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the Head of my Department or by his representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Mathematics

The University of British Columbia
2075 Wesbrook Place
Vancouver, Canada
V6T 1W5

Date June 30, 76
ABSTRACT

This study was motivated by the writer's observation that the provision of solved examples to students learning to devise computer algorithms did not assist and even seemed to hinder in the development of such skills. It was surmised that this might be due to a number of factors. The learner might be delayed in his development of the heuristics necessary to create algorithms using self-generated mediators. He might be misled in his expectation of the difficulty of performing such tasks independently. He might display rigidity (an Einstellung effect) in his later use of the techniques demonstrated by previously provided examples.

Grade nine students were assigned to two groups at random. Both groups were given a printed introduction to computer program writing in the BASIC language and were asked to solve two problems, an easy problem and a harder criterion problem. Before the problems were assigned one group was given a solved example which was very similar to the easy problem. The second group was given a short history of computers to read.

A Chi-square test was used to test each of the following hypotheses:

1. The first problem was easier than the second problem for all students.
2. The example helped the first group in doing the easy problem comparing the proportion of correct solutions to the easy problem in each group.

3. The second group had a higher proportion of correct solutions for the "hard" problem than the first group.

4. The second group had a higher proportion of correct solutions for the "hard" problem than the first group when only those students who correctly solved the first problem were considered.

The first, second, and fourth hypotheses were found to be significant beyond the .05 level.

The conclusion was drawn that the use of examples to teach algorithm development on the computer is at least sometimes inadvisable in that examples may hinder transfer of training from easy problems to harder problems and do not increase the numbers who can independently solve a harder problem. (This assumes that the independent solution of harder problems is the only instructional goal.) At best the provision of such examples may be a waste of time, at worst it may be a distraction.

It was felt that further research using a greater number and variety of examples, classified in some way, and using a variety of textual material is both warranted and desirable.
It was also felt that a test instrument could be devised which would identify those students who would most benefit from a course in algorithm development on the computer.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. THE PROBLEM</strong></td>
<td>1</td>
</tr>
<tr>
<td>Background</td>
<td>1</td>
</tr>
<tr>
<td>Purpose of the Study</td>
<td>5</td>
</tr>
<tr>
<td>Significance of the Study</td>
<td>5</td>
</tr>
<tr>
<td>Related Studies</td>
<td>8</td>
</tr>
<tr>
<td>Transfer</td>
<td>8</td>
</tr>
<tr>
<td>Einstellung</td>
<td>10</td>
</tr>
<tr>
<td>Algorithm Development</td>
<td>14</td>
</tr>
<tr>
<td>Statement of the Hypotheses</td>
<td>22</td>
</tr>
<tr>
<td><strong>II. THE DESIGN OF THE STUDY</strong></td>
<td>23</td>
</tr>
<tr>
<td>Introduction</td>
<td>23</td>
</tr>
<tr>
<td>The Instructional Device</td>
<td>23</td>
</tr>
<tr>
<td>The Experimental Device</td>
<td>23</td>
</tr>
<tr>
<td>The Control Device</td>
<td>23</td>
</tr>
<tr>
<td>The Test Instrument</td>
<td>24</td>
</tr>
<tr>
<td>The Pilot Studies</td>
<td>26</td>
</tr>
<tr>
<td>The Selection of Classes</td>
<td>26</td>
</tr>
<tr>
<td>Instruction of Teachers</td>
<td>27</td>
</tr>
<tr>
<td>Formation of Groups</td>
<td>28</td>
</tr>
<tr>
<td>Procedure</td>
<td>28</td>
</tr>
<tr>
<td>Apparatus</td>
<td>29</td>
</tr>
<tr>
<td>Statistical Analysis</td>
<td>30</td>
</tr>
<tr>
<td>Introduction</td>
<td>30</td>
</tr>
<tr>
<td>Chapter</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Data</td>
<td>30</td>
</tr>
<tr>
<td>Null Hypotheses</td>
<td>31</td>
</tr>
<tr>
<td>Statistical Treatment</td>
<td>32</td>
</tr>
<tr>
<td>III. ANALYSIS OF THE RESULTS</td>
<td></td>
</tr>
<tr>
<td>Summary of Data</td>
<td>34</td>
</tr>
<tr>
<td>Testing of the Hypotheses</td>
<td></td>
</tr>
<tr>
<td>Hypothesis One</td>
<td>34</td>
</tr>
<tr>
<td>Hypothesis Two</td>
<td>36</td>
</tr>
<tr>
<td>Hypothesis Three</td>
<td>37</td>
</tr>
<tr>
<td>Hypothesis Four</td>
<td>38</td>
</tr>
<tr>
<td>Conclusions</td>
<td>39</td>
</tr>
<tr>
<td>Discussion of Additional Data</td>
<td>39</td>
</tr>
<tr>
<td>IV. CONCLUSIONS AND IMPLICATIONS OF THE STUDY</td>
<td></td>
</tr>
<tr>
<td>Introduction</td>
<td>40</td>
</tr>
<tr>
<td>Discussion of the Conclusions</td>
<td>40</td>
</tr>
<tr>
<td>The Content of the Lesson</td>
<td>40</td>
</tr>
<tr>
<td>The Method of Instruction</td>
<td>41</td>
</tr>
<tr>
<td>Length of the Experiment</td>
<td>41</td>
</tr>
<tr>
<td>Group Makeup</td>
<td>41</td>
</tr>
<tr>
<td>Mechanics of Rigidity</td>
<td>41</td>
</tr>
<tr>
<td>Limitations of the Study</td>
<td>42</td>
</tr>
<tr>
<td>Suggestions for Future Studies</td>
<td>43</td>
</tr>
<tr>
<td>FOOTNOTES</td>
<td>45</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>48</td>
</tr>
<tr>
<td>CHAPTER</td>
<td>APPENDIX A - INSTRUCTIONAL DEVICE</td>
</tr>
<tr>
<td></td>
<td>APPENDIX B - EXPERIMENTAL DEVICE</td>
</tr>
<tr>
<td></td>
<td>APPENDIX C - CONTROL DEVICE</td>
</tr>
<tr>
<td></td>
<td>APPENDIX D - TEST INSTRUMENT</td>
</tr>
<tr>
<td></td>
<td>APPENDIX E - COMPUTER PRINTOUT</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. Summary of Data</td>
<td>34</td>
</tr>
<tr>
<td>II. Test of Hypothesis One</td>
<td>35</td>
</tr>
<tr>
<td>III. Test of Hypothesis One(a)</td>
<td>35</td>
</tr>
<tr>
<td>IV. Test of Hypothesis One(b)</td>
<td>35</td>
</tr>
<tr>
<td>V. Test of Hypothesis Two</td>
<td>36</td>
</tr>
<tr>
<td>VI. Test of Hypothesis Three</td>
<td>37</td>
</tr>
<tr>
<td>VII. Test of Hypothesis Four</td>
<td>38</td>
</tr>
<tr>
<td>FIGURE</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>1. Conceptual framework for Learning to Devise Algorithms</td>
<td>4</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENT

The author wishes to thank the teachers and students of West Vancouver schools for participation, my committee for anticipation, my wife for rehabilitation, and Professor B. Hicks (University of Illinois) for inspiration.
If students are provided with examples in order to teach them to devise their own algorithms, they may become rigid or stereotyped in their approach to algorithm development. Although teacher-provided examples may serve initially as mediators for the transfer of training, those students who become involved in systems work cannot continue to be provided with suitable mediating examples. Systems work is here defined to mean, "organizing collections of men, machines and methods." Such students must learn either to generate the examples themselves or to use some heuristic which is not dependent on examples. Each such student will use his skill in a different context from that in which it was first learned.

Personnel teaching in the author's mathematics computer laboratories have remarked that the provision of mediating examples in the early stages of teaching students to devise algorithms seemed to inhibit the later independent work of these students. It was stated that those students who were not provided with sample methods progressed more slowly at first but were more successful when later they had to transfer their algorithm development skills to more difficult individually chosen problems for which no examples could be provided.
Of eighteen students in Computer Science who were polled at the end of the course with respect to the value of sample programs, sixteen reported that examples inhibited progress or were useless.

This study investigates such claims. If these claims are valid the common practice of providing frequent examples in senior mathematics and computer science courses may be unsound.

The educational goal under consideration in this study was the ability to produce a plan or algorithm. This goal is called "Synthesis 5.2" in the Taxonomy of Educational Objectives. There was no interest in the particular algorithm itself (a number sequence generator), nor in the particular type of algorithm (a computer program in the BASIC language), but for simplicity of experimentation particular choices were made. The specific example introduced prior to the criterion tasks had to be capable of lower order mediation of transfer of training in algorithm development while at the same time inhibiting or at least not assisting higher order transfer by inducing a rigid approach to algorithm development (an Einstellung effect). Several pilot studies were undertaken in order to discover suitable subjects and materials.

The author's experiences with the idea of "computers in the whole school" and subsequent seminars on the same topic (at the University of Illinois, November 1971,
July 1974) had demonstrated the desirability of discovering the best means of leading a student so that, "His effort should yield a product...something that can be observed through one or more of the senses, and which is clearly more than the materials he began to work with." It is difficult to get students to produce a plan or algorithm for the computer. Bloom notes that "...current programs overemphasize activities in which the learner functions as a consumer and critic of ideas rather than those in which he functions as a producer." Although many students expect to learn largely by working examples through imitation of a teacher or text, it is difficult to see how any series of examples could lead a student to perform such sophisticated tasks as programming a computer to perceive sound and thereby operate a robot, or to play music. These tasks are accomplished by the students of Seymour Papert. Such sophisticated programming requires a great deal of synthesis, "the category of the cognitive domain which most clearly provides for creative behavior on the part of the learner." In the "production of a plan or proposed set of operations" to perform such tasks there clearly must be practice at the lower taxonometric levels of "comprehension", "application", and "analysis". This present experiment really operates only at these lower taxonomic levels, but where synthesis is the goal of instruction and if examples can be shown to be a hindrance even in low level preparation for that goal, then examples should be avoided at such levels of instruction.
The conceptual framework for learning to develop algorithms was as follows:

START:

Low order skills in terms of difficulty, complexity, and remoteness from past experience.
The background for this framework was derived largely from the work of R.M. Gagne and S.S. Lee. The background for the phenomenon of Einstellung rigidity was derived from A.S. and E.H. Luchins and K.M. Miller. The computer equipment and instruction techniques, as well as the problem itself, originated in the author's computer centres at Sentinel and Hillside Secondary Schools in West Vancouver, B.C.

PURPOSE OF THE STUDY

It was the author's purpose to investigate whether the provision of an example to students learning to devise their own computer algorithms resulted in a rigid and therefore unsuccessful approach to independent algorithm development.

SIGNIFICANCE OF THE STUDY

Gagne and Brown stated in 1961\(^1\)\(^2\) that "guided discovery" was most effective, "discovery" next most effective, and "rule and example" least effective in producing transfer of conceptual learning. It would seem a logical extension to the Gagne and Brown model if provision were made for judging the effectiveness of examples to mediate "guided discovery" or to contribute to the relative ineffectiveness of "rule and example". This paper may aid the study of the problem by determining if an example could inhibit or at least fail to assist higher order transfer of algorithm-building learning.
sets and yet be capable of lower order transfer.

If the guidance of discovery through teacher-provided mediating examples must be terminated at some point, the learner must either be able to intentionally develop mediators for himself or he must learn to do without mediators altogether. For example, if the learner becomes a biologist who investigates the relationship between the sizes of rabbit and fox populations, and if he wishes to produce a computer algorithm for the guidance of game management personnel, he must be able to relate concepts in ecology and statistics to skills in mathematics and computer programming. Calling upon his past experience in these fields he must develop a new algorithm which may be used by those not possessing his competencies. He must be able to synthesize relevant techniques in a flexible rather than in a stereotyped manner.

If examples are provided for low order tasks the learner may be rendered a disservice in several ways. He may be delayed in his development of the heuristics necessary to create algorithms using self-generated mediators. He may be misled in his expectation of task difficulty by early superficial successes and poor motivation might result from unaccustomed large increments in difficulty when mediation is withdrawn in advanced courses or at school leaving. He might display an Einstellung effect in his handling of data and techniques.
There would seem to be implications here for workers in programmed instruction, computer assisted instruction, and text-book writing as well as for mathematics and science classroom teachers. It is often assumed, it appears, that specific examples of a process being taught will always mediate transfer to higher order uses of the process (see for example, J. Walther, Computer Assisted Mathematics Program, Scott Foresman and Co., 1969). Even if it is claimed that examples are used only to aid in the explanation of terms and concepts and not to mediate their use, it may be that care must be exercised in the use of examples so that rigidity of mental processes is not produced as an undesirable side effect.

It may be that such topics as algorithm development in mathematics and computer programming should be taught in the same manner as the game of chess; that is, the moves of each piece are explained (the meaning of symbols, commands) and then the learner is simply allowed to play (devise algorithms, write programs). At a much later stage, after the learner has developed his own methods, he might safely undertake the study of special techniques by comparing his methods with those of experts. It might be safe to learn by example at this stage as there might be less chance of adopting a rigid approach to the task.
RELATED STUDIES

A foundation will be laid for the consideration of transfer of training in learning hierarchies by a discussion of the work of R.M. Gagne. The work of A.S. and E.H. Luchins and K.M. Miller will introduce the phenomenon of Einstellung. M. Wertheimer and J.M. Scandura are cited for their work on problem solving and algorithms; G.M. Haslerud, P. Suppes, T.A. Romberg and S.S. Lee for studies on guided discovery, item structure, cognitive individual differences and chaining cues respectively.

Transfer.

In 1949 Gagne\textsuperscript{13} studied the problem of measuring transfer. He concluded that the best means were; raw score, percentage improvement due to the contribution of the transferred task, percentage improvement during trials and presence or absence of transfer as measured by coefficients of correlation. He stated that other measures of the effect of transfer of training suffered from lack of compatibility of empiricism.

In 1961 he\textsuperscript{14} investigated the relationships between learning sets in knowledge acquisition and concluded that there was a positive transfer to each new learning set from relevant subordinate past learning and that the correlation between the mastery of this subordinate material and achievement
was highest at the highest level of his learning hierarchy. In 1961\textsuperscript{15} he also investigated the effects of variation in programming conceptual learning materials on learning problem solving as measured by the time required, hints required and score obtained. The problems used involved deriving the formula for the sum of terms in unfamiliar number series. Such formula writing may be considered as 'low level algorithm development.'

By 1962 Gagne\textsuperscript{16} was working with auto-instructional devices to define what he calls "productive learning". By this phrase he meant the kind of change in human behavior which permits the individual to perform successfully an entire class (or system) of specific tasks rather than one member of the class. At this time he stated that, "there are no instances of an individual who is able to perform what has been identified as a 'higher level' learning set, and who then shows himself unable to perform a 'lower level' learning set related to it." He further stated that the rate of attainment of learning sets in an hierarchy comes to depend on an increasing extent on the learning sets which have just previously been acquired and accordingly to a decreasing extent upon a basic factor or ability.

In 1963 Gage\textsuperscript{17} pointed out Gagne's questioning of the assumption that the best way to learn a performance is to practice that performance.
"In conditioning, classical or otherwise, one observes learning only after the animal has made the first response and one conceives of what is learned as either a response or an association terminating in a response, in either case established by practicing the response with reinforcement. Gagne challenged this on the grounds that the responses required do not have to be learned at all - they are already in the human's repertoire..."

"In training men to trouble shoot (Gagne 1962) complex equipment, there was no single task to be produced, rather it was the learning of an elaborate set of rules pertaining to the flow of signals through a complex circuit - another cognitive structure (system) that proves essential. Rather than response elicitation and reinforcement, as is implied by at least some interpretations of conditioning theory, the more important principles in training, in Gagne's view, deal with task analysis, intra task transfer, component task achievement, and sequencing."

That is, the realization of the structure was more important than the response elements.

Einstellung.

Luchins\textsuperscript{18} pointed out possible deleterious effects of habituated behavior. His method was as follows. Several problems, all solvable by one somewhat complex procedure, were presented in succession. The problems involved using three water jars of given capacity to produce a certain volume of water. Then a similar problem was given which could be solved by a more direct and simple method. This problem was called an "extinction problem". He found that most subjects persisted in attempting to use the complex method. Although the order of difficulty was reversed in the present study, and although
only two problems were presented, the idea of Einstellung rigidity is still present in that the student has the opportunity and the tendency to copy a technique rather than use a direct method. Luchins found that the tendency to copy the previous method of solution in subsequent problems was independent of educational level, age, and I.Q. He further found that attempts to prevent this mental rigidity or Einstellung effect were quite ineffective. He tried to prevent its occurrence by telling subjects to 'avoid blindness' after doing a certain problem and before doing the next. He made the problems more concrete, he provided facilities for experimentation, all to no avail. He observed that a habit of problem solution,

"ceases to be a tool discriminatedely applied, but becomes a procrustean bed to which the situation must conform; when, in a word, instead of the individual mastering the habit, the habit masters the individual."\(^{19}\)

Luchins recommended that learners must become accustomed to problems with too much data. They must be prevented from learning 'type exercises.' He observed that the usual classroom practice of teaching a specific topic and then assigning problems only on that specific topic contributes to the formation of Einstellung effects in problem solving. For example, after a lesson on two-digit multiplication the teacher might assign as a first exercise a question in subtraction. He surmised that speeded tests also contribute to an Einstellung
effect in that probably the most useful piece of information related to some problems is the fact that they can be done in so many minutes. The demand for speed causes blindness. Upon experimenting with changes in these usual classroom procedures he was told by the students, "I did what I was told to do", "you tricked us", "you taught us wrong."\(^{20}\)

Luchins further stated that if tests were interesting, if substantial mark allowances were made for method of attack, and if tests were presented as a method of helping students, there would be less mechanization of thinking fostered by them.

Miller\(^{21}\) in 1957 found results contradicting some of Luchins earlier tentative conclusions. He found that there is a significant (negative) relation between Einstellung rigidity or mechanization in problem solving and intelligence. He also found that when sub-samples of technical and modern school boys were matched for intelligence and compared, the Einstellung effect occurred significantly more often in the modern school group (who were drilled) than in the technical school students (who were trained to search for alternative methods). This result he discussed as a function of teaching methods and attitude to school. Miller used a single school to regulate some factors not controlled in the Luchin's study such as physical surroundings, teacher personality and socio-economic status of parents. The present writer has followed
Miller's lead in this and assumed that the distribution of intelligence and age into sub-groups was achieved by randomization. The Miller method of scoring (counting the number of control and extinction problems solved) was used and like Miller a pilot study was used to determine suitability of problem difficulty and $S$ grade level for the research at hand.

In the discussion of his experimental results Miller remarked,

"depending on what one considers the aim of education, it could be argued that in suiting teaching methods to the less able modern stream pupils, the more able are prevented from developing as flexibly as they otherwise might have done. On the other hand if it is considered desirable to train pupils to approach the solution of problems in a mechanical way, the above findings need cause no concern."

The present writer will comment upon this further in Chapter IV. It is sufficient at this point to comment that the selection of non-rigid students for training in independent high level algorithm development on computers, might be more productive than the selection of high I.Q. students. Although lack of rigidity tended to accompany high I.Q. the more specific and direct method of selection is more appropriate in screening individual talents.

Miller's comments on school morale and climate are particularly important to teachers considering the installation of computers and computer training programs similar to those
of the present writer. The present experiment was carried out in a school with a friendly climate but with a 'structured' academic approach to education. The school was in an upper socio-economic area.

Algorithm Development and Problem Solving.

Wertheimer described problem solving as a search for structure which is like the search for the reversed image in an optical illusion. Perhaps some structures (such as new algorithms) must be sought by the individual from the beginning without the aid of the mediators mentioned in the conceptual framework (figure 1).

Scandura related some effects of algorithm learning and problem solving. It should be noted that this is not identical with the central issue of this paper, algorithm development, however Scandura's results are useful in that they show what algorithms are, how they can be generalized and how they relate to higher order transfer in problem solving.

Scandura described algorithms as follows;

"Algorithms exist for solving many types of problems. The step-by-step computational procedures used in arithmetic perhaps provide the most familiar examples, but algorithms are also used in dealing with all sorts of practical and theoretical problems - from 'trouble-shooting' to mathematical physics. A common feature of such procedures is that they can be applied mechanically without understanding."
As a computer has no 'understanding' and is truly 'mechanical' it obviously solves problems by means of algorithms. The algorithms developed for computer use usually can be extended in application to human use and so Scandura's statement would seem to support the belief that there exists a continuing need for algorithm developers and algorithm users, who are not necessarily distinct persons. Certainly the person who originally devised the algorithm should understand the theory of its operation completely.

Scandura continues;

"Many educators and subject matter specialists would maintain that algorithms are limited in their usefulness to specific problem situations and that transfer to variants of the original problems requires an understanding of underlying principles. The fact that understanding is typically defined in terms of performance on some transfer task, however, poses a problem for researching such conjectures. To operationally define understanding in terms of problem solving transfer would clearly lead to circularity. One way to overcome this difficulty is to operationally define 'understanding' in terms of the amount of information presented."

The present study, as will be seen, varied the amount of information to determine the degree of understanding.

In a series of three experiments Scanduras demonstrated that:

a) Successful problem solving does not necessarily depend on an understanding of the problem involved. (This lends
support to the modern notion of 'systems' in which symbol manipulation can often be substituted for understanding of the problem).

b) Transfer does not depend on 'understanding'. That is, it is possible that subjects detect relationships and cues among algorithms and are able to modify them according to syntactic constraints present between the algorithm and the individual problem characteristics. (This would seem to be an essential skill for a systems programmer).

c) It is possible that, instead of asking, like Gagne, 'What would S need to know in order to do this task?', one could be concerned with the problem, 'Could information facilitate problem solving?'. In other words, the structured relationships could be of prime importance.

d) It is feasible to predict student problem solving performance on a given topic by subjectively analysing structural relationships between the performance criterion and the information presented.

e) Mere presentation of subordinate material is not always sufficient to ensure subsequent learning when the terms (i.e. symbols and words) denoting the subordinate notions are used to describe the higher order material. That is, Gagne was concerned with skills which are more readily taught and not
with the presentation of continuous discourse in terms of subordinate associations, concepts and principles.

f) There are at least some conditions under which prerequisite practice has a greater effect on problem solving performance immediately after learning than an equal amount of practice time at the criterion level. Scandura notes that the problem of glossing over preliminary topics so as to spend more time on 'main ideas', particularly when the preliminary topics are relatively unfamiliar to the students, is a very real problem in many college classrooms, particularly in the technical and scientific areas. (This lends some support to the notion that the provision of mediators in preliminary material may be a means of glossing over necessary practice in algorithm development).

g) Problem solving is improved by the pre-experimental availability of prerequisite material, even after repeated re-introduction of criterion level materials, hints and practice in problem solving, when the prerequisite material was presented prior to the criterion material, and by prerequisite practice only when the prerequisite material came first.

In 1967\textsuperscript{30,31} Scandura added these conclusions;

h) Knowing how to use an algorithm is different
from knowing \textit{when} to use it. There is a tendency to try to solve a group of problems all in the same way. (cf. Einstellung).

i) The more \textit{general} the introductory material and examples are the better use will be made of them.

The present author would comment that solved examples are probably the least general form of instruction. To remember a rule and an example may be simply taking the path of least resistance. In examining a solved example at a prerequisite level which is intended to transfer generalized skills to higher order tasks the level of transfer may be too low - that is, it may be too specific. The $S$ might simply see an exemplar of a problem type rather than of a concept. The effect of cues learned may then have induced an Einstellung effect. Scandura noted that the results of his experiment, however, were not unequivocable as the effect may depend upon the material (groups and topology) used. The same may be said for the results of the present experiment.

Scandura used an example which assisted $S$s in solving the first problem presented in much the same way as the present E. Scandura, however, provided the helpful example to all $S$s and so was unable to discriminate between the effect of providing or not providing the example. The present study looked at this question and thereby reinforced and expanded Scandura's statements concerning the differences in achievement
on extinction problems between those who are and those who are not "rule-users".

Scandura also set a precedent for the use of historical material as a filler for the control group and he confirmed experimentally that this should not confound research into transfer of training.

Haslerud in reviewing the work of Katona (who maintained that teaching by examples was best) and Craig (who found that the more guidance provided the more efficient the discovery) countered their work with an experiment dealing with coding problems which showed that unless a principle was derived by the learner (rather than stated and demonstrated through examples) there was no transfer of the principle to other problems. One would expect to find similar results from experiments concerning problem solving in computer programming.

Suppes stated that the learning of simple mathematical concepts is more closely associated with an 'all or none' assumption than by a simple incremental assumption: That is, research which attempted to show the process of such learning may simply have shown the probability $p$ that the subject had reached the conditioned state (when $p=1$). This may throw doubt upon the relevance of some statements concerning 'transfer of training' but it helped to explain those studies which showed 'lower order transfer' with no
accompanying 'higher order transfer'.

Suppes\textsuperscript{34}, like Gagne, worked on the creation of a model which can predict from an item's structure the process a subject must go through in finding a correct response. It should be noted that in this paper a 'correct response' was not equated with 'learning a concept'.

Romberg\textsuperscript{35}, in contrast to Suppes, stated that,

"...it seems plausible that many instructional procedures in mathematics could be guided by appropriate utilization of information on cognitive individual differences, but this is not the case. As individualization of instruction and computer management become a reality, aptitude and ability data should become extremely useful. Perhaps in the next decade cracks in the iron curtain will appear."

Among the aptitudes to which Romberg refers should be included the ability to resist Einstellung rigidity.

Lee\textsuperscript{36,37} continues to investigate the effect of chaining cues (mediators) on learning 'complex conceptual rules' such as labelling objects of various sizes colours and shapes. He and Gagne seem to believe in the existence of unique bodies of component learning which are necessary for the transfer from lower to higher level concept. However chaining is only possible when all the links in the chain can be found. In today's interdisciplinary systems, design cues may be hard to find and the identification of component learning for tomorrow's task impossible. Training in 'looking
outside the problem' and 'avoiding blindness' may be what is required for at least some members of the population.
STATEMENT OF THE HYPOTHESIS

It is hypothesized that a group of students who succeed on a low order computer algorithm development task after the provision of a specific example will have proportionally fewer successes on a higher order task than will a group of students who succeeded on the low order task without the use of the specific example.

In particular:

1. The first problem is easier than the second problem for all students.

2. The example helps the first group in doing the "easy" problem comparing the proportion of correct solutions to the easy problem in both groups.

3. The second group have a higher proportion of correct solutions for the "hard" problem than the first group.

4. The second group have a higher proportion of correct solutions for the "hard" problem than the first group when only those students who correctly solve the first problem are considered.
CHAPTER II

THE DESIGN OF THE STUDY

INTRODUCTION

The following four sets of material were first prepared.

The Instructional Device.

The instructional device consisted of a three page duplicated booklet which contained, "Directions to Teachers and Students", and "The Lesson". (cf. Appendix A) "The Lesson" provided instruction in the use of the BASIC computer language. Printed materials were used in order to control the teacher variable.

The Experimental Device.

The experimental device consisted of a single duplicated page showing a solved example of a program to print the odd numbers. (cf. Appendix B) The intent of this example was to make the 'easy' task very easy for the experimental group.

The Control Device.

The control device consisted of a single duplicated page giving a brief history of computers. (cf. Appendix C) This was to be used by the control group while the experimental
group used the experimental device.

The Test Instrument.

The test instrument consisted of a single duplicated page which identified the student, his group (A for experimental, B for control), and assigned two tasks. (cf. Appendix D) Task one (the 'easy' task) was to program the computer to print the even numbers. Task two (the 'difficult' task) was to program the computer to print the Fibonacci numbers.

The reasons for having the algorithm development skills developed on a computer were:

1. Computer language could be used to develop algorithms.

2. It would be possible to find Ss having no previous experience with computer languages thereby helping to control the effect of past learning experiences.

3. As four new terms (computer commands) were to be introduced the confounding effect of previously learned vocabulary was reduced. At the same time complete algorithms could be programmed using only these four new commands (LET, PRINT, END, GOTO).

4. As the learners would never have been instructed in computer languages by any other instructional mode the confounding effect of previous learning methods was reduced.
5. The evaluation of success could be accomplished quickly by an 'unbiased' computer with little of any kind of teacher or experimenter effect.

6. The whole subject of educational uses of computers was of particular interest to the experimenter.
THE PILOT STUDIES

The materials described above were used in two pilot studies to determine their suitability as to grade level, vocabulary level, time limits and problem difficulty. The pilot studies were executed in the same manner as the main experiment which is described later. It was found in the first pilot study with grade six students of several West Vancouver elementary schools that the grade six level was too low to produce a sufficient number of correct programs. In the second pilot study which used grade nine students from a single West Vancouver secondary school, it was found that revision in the vocabulary level of both the instructional device and the test instrument were required. These needed revisions were undertaken and the final versions which appear in the appendices were produced. Time limits as described later were set on the basis of the second pilot study so that no student required more time.

THE SELECTION OF CLASSES

Classes which were to be used for the experiment were those classes at a secondary school in West Vancouver, B.C. which were made available to the E by their mathematics teachers. These teachers agreed to attend a meeting to learn about the experiment and then to administer the materials under the guidance of the E. A mutually agreeable hour was found so that
all classes could use the materials at the same time (with their own mathematics teacher in attendance) to ensure that there would be no interaction between classes.

INSTRUCTION OF TEACHERS

The teachers whose classes were to be used for the experiment attended a meeting one week prior to the experimental session at which the E explained the purpose of the experiment. The E also explained that those students who admitted to previous experience with computers were not to be used in the experiment. Each teacher was provided with a table of random numbers and shown how to assign students to group A or group B. Time limits were explained to the teachers and the necessity of preventing any talking or copying by the students was emphasized. It was explained that the E would go from room to room while the experimental session was in progress to ensure that groups were formed and treated in a uniform manner (the four classrooms to be used were in the same wing of the building). The directions included in the instructional device were discussed and reasons given for these directions. The teachers were told that they were to allow ten minutes for the formation of groups and distribution (face downward) of "The Lesson" and the Group A or Group B page to each student. They were told to allow fifteen minutes for the reading of this material, five minutes for distributing (face downward) the test instrument,
ten minutes for working on task one, and fifteen minutes for working on task two (the teacher ensured, visually, that no student worked on task one when told to go on to task two). It was explained that after the bell sounded signalling the end of the test period the teacher was to collect all materials and ask students to report whether they would have liked more time or verbal explanations, and whether they would like to meet with the E to discuss the materials. The teachers were asked to report the answers to these questions to the E.

FORMATION OF THE GROUPS

The population was formed as described above from members of four grade nine classes which were made available to the E by their mathematics teachers. Students who had previous computer experience were rejected from the sample. Each student was randomly assigned to group A or group B using a table of random numbers. Ninety-two subjects were used; forty-nine in group A and forty-three in group B.

PROCEDURE

On the day of the experiment the E was an observer in the four experimental classrooms while the teachers handled the materials as described above. E ensured that the table of random numbers was used correctly and that the correct materials were distributed, the correct times kept, and that there was no interaction between Ss.
All teachers reported that their students found the time limits adequate and that no student expressed a desire for verbal explanation. Three of the four teachers reported that their students expressed a desire to meet with the E and discuss the materials. The E visited these classes during their regular mathematics classes during the week following the experimental session and answered questions directed to him by the students.

The students' task performances were judged both through computer operation upon their programs and by objective observations of student programming errors. The criterion for task success was the actual production by the computer, using Ss programs, of the required number sequences. In this way the marker variable was controlled.

APPARATUS

The apparatus consisted initially of a Hewlett-Packard 2007 system computer with card reader and BASIC software. The final version of the experiment used a Hewlett-Packard 9830 computer with high speed printer, optical mark card reader and BASIC software.
STATISTICAL ANALYSIS

Introduction.

A Chi-squared test was used to investigate the difference in:

1. The proportion of successes on the easy task as compared to the second task for all Ss and within each group.
2. The proportion of successes on the easy task in group A as compared to group B.
3. The proportion of successes on the difficult task in group A as compared to group B.
4. The proportion of successes on the difficult task in group A as compared to those in group B who were successful on the easy task.

Data.

For each student the two tasks were marked as successes or failures on the basis of whether they produced the correct computer print-out. In addition a subjective decision was made on whether each group A subject attempted to use the method of the provided example and of the first task in order to do the second task. A subjective decision was made on whether each group B subject attempted to use the method of task one in order to do the second task.
Null Hypotheses.

H 1. There is no significant difference in the proportion of Ss who succeeded on the first task compared to the proportion of those who succeeded on the second task.

H 1a. There is no significant difference in the proportion of Ss in group A who succeeded on the first task compared to the proportion of those who succeeded on the second task.

H 1b. There is no significant difference in the proportion of Ss in group B who succeeded on the first task as compared to the proportion of those who succeeded on the second task.

H 2. There is no significant difference in the proportion of Ss in group A who succeeded on the first task compared to the proportion of Ss in group B who succeeded on the first task.

H 3. There is no significant difference in the proportion of Ss in group A who succeeded on the second task compared to the proportion of Ss in group B who succeeded on the second task.

H 4. There is no significant difference in the proportion of Ss in group A who succeeded on the second task compared to the proportion of Ss in group B who succeeded on the second task if only those subjects who were successful on the first task were examined.
Statistical Treatment.

The following chi-squared statistics were calculated with one degree of freedom and were tested at the .05 level of significance. A1 indicated the number of successes of Ss in group A on task 1. A2, B1, B2 were interpreted similarly. A*2 and B*2 indicated the number of successes in groups A and B on task 2 of those Ss who succeeded on task 1. 'N' referred to the total number of Ss involved in the experiment. '#Correct' referred to the number of successes in each category.
<table>
<thead>
<tr>
<th></th>
<th>#Correct</th>
<th>#Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>A1+B1</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>A2+B2</td>
<td>G</td>
</tr>
<tr>
<td>H1a</td>
<td>A1</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>A2</td>
<td>G</td>
</tr>
<tr>
<td>H1b</td>
<td>B1</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>B2</td>
<td>G</td>
</tr>
<tr>
<td>H2</td>
<td>A1</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>B1</td>
<td>G</td>
</tr>
<tr>
<td>H3</td>
<td>A2</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>B2</td>
<td>G</td>
</tr>
<tr>
<td>H4</td>
<td>A*2</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>B*2</td>
<td>G</td>
</tr>
</tbody>
</table>

In each case the computer calculated the chi-squared statistic according to the formula

\[ \text{CHI SQUARED} = \frac{N (EH - FG)^2}{(E+F)(G+H)(E+G)(F+H)} \]

(cf. Appendix E for the computer statistical treatment)
CHAPTER III
ANALYSIS OF THE RESULTS

The following table summarizes the data obtained.

TABLE I
SUMMARY OF DATA

<table>
<thead>
<tr>
<th>Number of Ss who:</th>
<th>Group A</th>
<th>Group B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Had neither task correct.</td>
<td>6</td>
<td>20</td>
</tr>
<tr>
<td>Task 1 correct only.</td>
<td>31</td>
<td>9</td>
</tr>
<tr>
<td>Task 2 correct only.</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Tasks 1&amp;2 correct.</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>One or both tasks correct.</td>
<td>43</td>
<td>23</td>
</tr>
<tr>
<td>Task 1 correct.</td>
<td>42</td>
<td>23</td>
</tr>
<tr>
<td>Task 2 correct.</td>
<td>12</td>
<td>14</td>
</tr>
<tr>
<td>Showed Einstellung effect.</td>
<td>14</td>
<td>1</td>
</tr>
<tr>
<td><strong>Number</strong></td>
<td><strong>49</strong></td>
<td><strong>43</strong></td>
</tr>
</tbody>
</table>

TESTING OF THE HYPOTHESES

Hypothesis one.

Hypothesis one was that there would be no significant difference in the proportion of subjects who succeeded on the first task as compared to the proportion of those who succeeded on the second task. Hypotheses 1a and 1b dealt with each group of Ss separately. The following tables summarize the results obtained:
### TABLE II

**TEST OF HYPOTHESIS ONE**

<table>
<thead>
<tr>
<th></th>
<th># Correct</th>
<th># Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Al+B1</td>
<td>65 (71%)</td>
<td>27 (29%)</td>
</tr>
<tr>
<td>A2+B2</td>
<td>26 (28%)</td>
<td>66 (72%)</td>
</tr>
</tbody>
</table>

Degrees of freedom = 1, Chi-square = 33.069

### TABLE III

**TEST OF HYPOTHESIS ONE(a)**

<table>
<thead>
<tr>
<th></th>
<th># Correct</th>
<th># Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>42 (86%)</td>
<td>7 (14%)</td>
</tr>
<tr>
<td>A2</td>
<td>12 (24%)</td>
<td>37 (76%)</td>
</tr>
</tbody>
</table>

Degrees of freedom = 1, Chi-square = 37.121

### TABLE IV

**TEST OF HYPOTHESIS ONE(b)**

<table>
<thead>
<tr>
<th></th>
<th># Correct</th>
<th># Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>23 (53%)</td>
<td>20 (47%)</td>
</tr>
<tr>
<td>B2</td>
<td>14 (33%)</td>
<td>29 (67%)</td>
</tr>
</tbody>
</table>

Degrees of freedom = 1, Chi-square = 3.842

The critical value of Chi squared at the .05 level of significance with one degree of freedom is 3.841

Since all three values of Chi-squared obtained exceeded the critical value, the Null hypothesis was rejected and it was concluded that there was a significant difference in the proportion of subjects who succeeded on the first task.
as compared to the second task. This result was obtained even with group B who had no example to assist them in doing the first task. It may therefore be concluded that the first task was found to be easier and is therefore a 'lower order task' in computer algorithm development.

**Hypothesis two.**

Hypothesis two was that there would be no significant difference in the proportion of Ss in group A who succeeded on the first task compared to the proportion of Ss in group B who succeeded on the first task. The table below summarizes the results obtained:

**TABLE V**

<table>
<thead>
<tr>
<th></th>
<th># Correct</th>
<th># Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>H2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Al</td>
<td>42 (86%)</td>
<td>7 (14%)</td>
</tr>
<tr>
<td>Bl</td>
<td>23 (53%)</td>
<td>20 (47%)</td>
</tr>
</tbody>
</table>

Degrees of freedom = 1, Chi-square = 11.471

Since the chi-squared value obtained exceeded the critical value the null hypothesis was rejected and it was concluded that there was a significant difference in the proportion of Ss in each group who succeeded on the first task.

It may therefore be concluded that the solved example provided to group A assisted them in performing the first task.
by acting as a mediator between the instructional material and the first task.

**Hypothesis three.**

Hypothesis three was that there would be no significant difference in the proportion of Ss in group A who succeeded on the second task compared to the proportion of Ss in group B who succeeded on the second task. The table below summarizes the results obtained:

**TABLEA1VI**

**TEST OF HYPOTHESIS THREE**

<table>
<thead>
<tr>
<th></th>
<th># Correct</th>
<th># Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2</td>
<td>12 (24%)</td>
<td>37 (76%)</td>
</tr>
<tr>
<td>B2</td>
<td>14 (33%)</td>
<td>29 (67%)</td>
</tr>
</tbody>
</table>

Degrees of freedom = 1, Chi-square = .735

Since the chi-squared value obtained did not exceed the critical value it was concluded that there was no statistically significant difference between the groups in regard to the proportion who succeeded on the second task. It cannot therefore be concluded that Ss in group A as a group were hindered by the example in performing the second task. Evidently the example was neither a hindrance nor a help to the whole of group A on the second task.
Hypothesis four.

Hypothesis four was that there would be no significant difference in the proportion of Ss of group A who succeeded on the second task compared to the proportion of Ss in group B who succeeded on the second task if only those Ss who were successful on the first task were examined. The table below summarizes the results obtained:

**TABLE VII**

**TEST OF HYPOTHESIS FOUR**

<table>
<thead>
<tr>
<th></th>
<th># Correct</th>
<th># Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>A*2</td>
<td>11 (26%)</td>
<td>31 (74%)</td>
</tr>
<tr>
<td>B*2</td>
<td>14 (61%)</td>
<td>9 (39%)</td>
</tr>
</tbody>
</table>

Degrees of freedom = 1, Chi-squared = 7.551

Since the chi-squared value obtained exceeded the critical value the null hypothesis was rejected and it was concluded that there was a significant difference in the proportion of Ss in the two groups who succeeded on the second task if only those who were successful on the first task were examined. It may therefore be concluded that the solved example hindered any transfer which might have occurred from the first task to the second task for those who successfully completed the first task.
CONCLUSIONS

Considering all four hypotheses together it may be concluded that the provision of an example which mediated transfer of skills to an easy task made no difference to the number of correct solutions on the harder, criterion task, but the solution of the easier problem (when a solved example was provided) did hinder the further utilization (transfer) of skills acquired in performing the first task.

DISCUSSION OF ADDITIONAL DATA

The above results are confirmed by the subjective scrutiny of unsuccessful S_ programs. Fourteen S_s in group A attempted to use the method of solution demonstrated in the example and appropriate to the first task. Only one S in group B attempted such a method.
CHAPTER IV

CONCLUSIONS AND IMPLICATIONS OF THE STUDY

INTRODUCTION

The central hypothesis of this study was that the use of a specific example which would mediate the transfer of skills from instruction to the solution of an easy problem would also inhibit the transfer of those same skills from an easy problem to a more difficult problem. That is, the student was not unsuccessful because of exposure to the easy task, but because of his solution of the task using a solved example. As significant results were obtained it was necessary to consider possible reasons.

DISCUSSION OF THE CONCLUSIONS

The Content of the Lesson.

In choosing the material for instruction the experimenter used pilot studies to assist him to choose a suitable reading and grade level. It may have been that the particular choices of written material and grade level as well as the interest value of the subject matter influenced the results. However it was the purpose of the study only to find if rigidity could be induced by the use of a specific example. Of course other examples were used in the instruction common to both experimental groups and the effects of these examples should be considered in further research.
The Method of Instruction.

For purposes of experimentation written directions, lessons, examples, and task selection were used. It may be that the results of this study reflect Ss reaction to written material. Such techniques as oral discussion and computer demonstration might have produced different results due to differences in educational climate, motivation, and expectation of success.

Length of the Experiment.

Although pilot studies and observations during experimentation confirmed that no S wanted more time, it is possible that the one hour long experience induced fatigue in some Ss and thereby influenced the results.

Group Makeup.

Although random assignment to groups was necessary in order to have equivalent groups and although it was made obvious that assignment to groups was random, there may have been a curiosity about the materials others were receiving which distracted some Ss.

Mechanics of Rigidity.

No conclusions have been formed as to the possible causes of rigidity discussed in the introduction: Whether an
original predisposition to rigidity is innate or learned, whether the learner is delayed by examples in his development of necessary heuristics, misled in his expectation of task difficulty or hindered in some other way. These questions are beyond the scope of this paper. It has been demonstrated, however, that rigidity can be induced in at least some Ss by the use of an example.

LIMITATIONS OF THE STUDY

Some limitations of the study have already been implied in the above discussions of the conclusions with respect to the lesson content, method of instruction, length of the experiment and group makeup. There are also of course the usual limitations of the assumption of randomization of intelligence. A further limitation is that all the Ss in the final experiment came from the same school. This school is in an upper socio-economic district and has a history of educational innovation. For the purposes of the question at hand, whether rigidity can be induced, the choice of a single school (with its particular socio-economic status and I.Q. range) was ideal in that it controlled a number of variables. For the purpose of generalizing the results of this study further, especially for text-book writers and teacher educators, the limitations are great. Further studies for generalization should be conducted in as wide a variety of schools as possible.
SUGGESTIONS FOR FUTURE STUDIES

Scandura demonstrated the feasibility of predicting problem solving performance by subjectively analysing structural relationships between the criterion and the information presented. In the present study a structural relationship involving copying was demonstrated if all printed materials up to the criterion level task were considered as 'information presented'. This structure was inappropriate at the final criterion level, yet it would seem to the author that precisely this structure is presented to students using many text books in high school mathematics and computer science. Typically the introduction to a topic is followed by three or four specific solved examples and then by a set of exercises. The student who uses these examples as generalizations obtains mediation of transfer from what he has read to the exercises which he must now do. The student who attempts to apply the methods of the examples too rigidly to the exercises fails on the more difficult exercises which, in higher level courses, may often be considered the true criterion level. Indeed, the first few problems in the exercise are frequently intended as no more than drill and practice in the use of vocabulary, definitions, and notation. Luchins and Miller have already demonstrated that some students have a predisposition to this sort of Einstellung rigidity and the present study shows that the example may reinforce this predisposition (if Einstellung
rigidity is thought of as a learned behavior). This will hinder what Gagne calls productive learning. It seems therefore that further research should be undertaken which investigates rigidity using a greater variety of examples, classified in some way, and a greater variety of textual material which contains effective warnings to avoid blindness. Such research should use larger blocks of instruction, using a whole text book topic and taking cognizance of Scandura's caveat that Ss will continue to respond in a rigid way unless confronted with feedback which indicates that 'the rule has changed'.

If it is felt that Einstellung rigidity is somehow a function of personality, then it would be appropriate to construct instruments which would identify which students would most benefit from a course in such disciplines as algorithm development on the computer. It could perhaps be shown that although there is a negative correlation between intelligence and Einstellung rigidity it is possible to construct specific tests for rigidity which are better for the purpose than intelligence tests.
Algorithm - "A prescribed set of well defined rules or processes for the solution of a problem in a finite number of steps, e.g. a full statement of an arithmetic procedure for evaluating sin x to a stated precision." American Standard Vocabulary for Information Processing (American Standards Association, 1966) p.9. The term "program" was not used as it might imply a direct translation into computer language or simply the revision of existing programs.

Mediator - a bridge for the transfer of training whereby relevant material and skills are recalled.

Systems work - "Organizing collections of men, machines and methods." American Standard Vocabulary, op.cit., p.26. Students in the author's grade eleven computer science course undertake the production of packages for such tasks as; school scheduling, reporting, attendance, accounting, computer assisted instruction, chemistry laboratory report marking, analysis of functions, graphing of conics, sunrise prediction, and genetic analysis.


Ibid., p.166.

Seymour Papert, "Teaching Children Thinking;" (memo LOGO Laboratory, 545 Technology Square, Cambridge Mass.).

Bloom, op.cit., p.162.


Ibid., p.91.


Ibid., p.134.


Ibid., p.1.

op.cit.

Ibid., p.1.


BIBLIOGRAPHY


Papert, Seymour. Teaching Children Thinking, Logo Laboratory, 1967.


APPENDICES
APPENDIX A

THE INSTRUCTIONAL DEVICE
DIRECTIONS TO TEACHERS AND STUDENTS

1. Hand out "THE LESSON" (these three pages)* to every student.

2. Hand out "GROUP A" page to half the students (approximately) and "GROUP B" page to the other half using a RANDOM process of distribution.

3. Students have 15 minutes to read these three pages plus their extra GROUP A or B page.

4. Students have 10 minutes to work on Problem 1.

5. Students have the rest of the period to work on Problem 2.

NOTE: Please feel free to turn back to the Lesson at any time but please do not work on Problem 1 when asked to work on Problem 2.

THE LESSON

Groups A & B

A program is a list of instructions for a computer. You are now going to learn how to program a computer. The list of instructions that makes up a computer program must be written in a special language. We will use the special computer language called BASIC. After you have studied the language called BASIC you will write two programs of your own. I will then put your programs into the computer to see if the computer 'understands' you.

THE BASIC LANGUAGE

The computer must have a name for every number. This name must be a letter of the alphabet. A statement which gives a name to a number must begin with the word LET. For example, if we wish to give the name X to the number 9, we will write this statement.

*Appendix note: Original was on three 8½ * 14 pages.
for the computer:

    LET X = 9

If we wish to give the name B to the number 4, we will write the statement:

    LET B = 4

Now, if we wish to have the computer add the number called X to the number called B, we must give a name to the answer. Let us choose A (although any letter except X and B would do just as well). We write the statement:

    LET A = X + B

Notice that the name or letter which is getting a new value must be written on the left of the equal sign. The computer must already know the numbers belonging to any letter name on the right of an equal sign. The computer obeys your program instructions or statements one at a time, starting with the first statement.

A statement which tells the computer to print an answer must begin with the word, PRINT. For example, if we wish to print the value of A, calculated before, we would give the statement:

    PRINT A

The last statement in a program must be the single word END.

Each statement must be written on a single line of your program. The computer will obey your statements in the same order you write them. We now know enough computer statements to write a computer program.
Here is our program:

```
LET X = 9
LET B = 4
LET A = X + B
PRINT A
END
```

When I put this program into the computer it will print only what we told it to PRINT. This is what it will print on the page.

![13]

We have a special reason for numbering each statement of our program. You will see the reason later. Our program now looks like this:

1. LET X = 9
2. LET B = 4
3. LET A = X + B
4. PRINT A
5. END

The reason we number each statement is that we sometimes wish the computer to go back to a previous statement. For this we use the statement beginning with the words GO TO. If we wish the computer to return to statement number 1 we write:

```
GO TO 1
```

For example, if we wish the computer to do the previous program again and again, we would write this program:

```
1. LET X = 9
2. LET B = 4
3. LET A = X + B
4. PRINT A
5. GO TO 1
6. END
```
This is what the computer would print:

13
13
13
13

The computer will never stop, of course, until I push a HALT button. It will do statement 1, then statements 2, 3, 4 then at statement 5 it will return to do statements 1, 2, 3, 4 and so on all over again. As you might guess, the computer will never get to step 6 in our example, but this step must be in the program anyway.

Of course we do not usually want the computer to print the same number again and again. Suppose we wanted the computer to use a value of 9 for X, but we wanted B to start with a value of 4 and then to grow 6 greater each time the computer did a calculation. We could then write the following program:

1. LET X = 9
2. LET B = 4
3. LET A = X + B
4. PRINT A
5. LET B = B + 6
6. GO TO 3
7. END

Statement 5 means, let the new value of B equal 6 more than the old value of B. This is what the computer would print.

13
19
25
31
If we had wanted the computer to add 9 to the value of B, for each calculation, instead of 6, statement 5 could have been:

5. LET B = B + X

This is what the computer would have printed:

13
22
31
40

If we had wanted the first value of B to be printed before the other number, so that the computer would print:

4
13
22
31
40

our program would have looked like this:

1. LET X = 9
2. LET B = 4
3. PRINT B
4. LET A = X + B
5. PRINT A
6. LET B = B + X
7. GO TO 4
8. END
APPENDIX B

THE EXPERIMENTAL DEVICE
GROUP A

A FINAL EXAMPLE OF A PROGRAM.

Here is a program that will print the odd numbers like this without ending:

1  3  5  7  9  11

1. LET X = 1
2. PRINT X
3. LET X = X + 2
4. PRINT X
5. GO TO 3
6. END
APPENDIX C

THE CONTROL DEVICE
Here is a list of names and dates which have to do with computers.

1. Muller 1786
2. Babbage 1820
3. Hollerith 1889
4. Aitken 1937
5. Bell Telephone 1938
6. Eckert 1949

1. Proposed the idea of a calculating machine.
2. Designed a calculating machine.
3. Developed a data processing machine.
4. Built the Mark I computer.
5. Built electromagnetic calculators.
6. Used electronic techniques for computation.
APPENDIX D

THE TEST INSTRUMENT
IMPORTANT: Make sure you put your Group in the blank below. You will find whether you are group A or group B by looking at the top of the other single page you were given.

GROUP.... NAME......................

SCHOOL......................

GROUP A & B

USE SCRAP PAPER FOR ROUGH WORK, THEN COPY YOUR FINAL ANSWERS ON THIS PAGE

Problem 1: Write a computer program which will make the computer print the even numbers, like this, without ending.

```
  2
  4
  6
  8
 10
```

Problem 2: Write a computer program which will make the computer print the following sequence of numbers, without ending.

```
  1  2  3  5  8 13 21
```

Note that if you let \( X = 0 \) and \( Y = 1 \) then the first number in the sequence is \( F = X + Y \). If you now let \( X \) be the same as \( Y \) was, and if you let \( Y \) be the same as \( F \) was, you get the second number in the sequence \( F = X + Y \) and so on.
APPENDIX E

THE COMPUTER PRINTOUT
PROGRAM

10 REM; CHI SQUARE FOR TWO BY TWO ARRAYS
20 PRINT "CORRECT", "INCORRECT"
30 PRINT "---------------
40 READ R, K
50 IF R#2 AND K#2 THEN 270
60 MAT A = ZER
70 REM; MATRIX OF DATA IS CLEARED
80 LET N=C=0
90 REM; N=GRAND TOTAL, C= CHI SQUARED
100 FOR E=1 TO R
110 REM; E IS CURRENT ROW
120 FOR D=1 TO K
130 REM; D IS CURRENT COLUMN
140 READ A(E,D)
150 REM; READ DATA FOR ROW E AND COLUMN D
160 PRINT A(E,D),
170 LET N=N+A(E,D)
180 REM; ADD EACH TOTAL TO GRAND TOTAL
190 LET A(E,K+1)=A(E,K+1)+A(E,D)
200 LET A(R+1,D)=A(R+1,D)+A(E,D)
210 REM; STEPS 190,200 ADD ROW & COLUMN TOTALS
220 NEXT D
230 PRINT
240 NEXT E
250 PRINT "DEGREES OF FREEDOM = "(R-1)*(K-1),
260 GOTO 290
270 PRINT "SWITCH TO PROGRAM FOR R BY K CHI SQUARE."
280 GOTO 410
290 LET T=N*((ABS(A(1,1)*A(2,2)-A(2,1)*A(1,2)))^2)
300 LET C=T/(A(1,3)*A(2,3)*A(3,1)*A(3,2))
310 PRINT "CHI SQUARE = "C
320 PRINT "================================================
330 REM; THIS IS CHI SQUARE ACCORDING TO VAN DALEN PAGE 413.
340 GOTO 40
350 DATA 2,2,65,27,26,66
360 DATA 2,2,42,7,12,37
370 DATA 2,2,23,20,14,29
380 DATA 2,2,42,7,23,20
390 DATA 2,2,12,37,14,29
400 DATA 2,2,11,31,14,9
410 END
CORRECT     INCORRECT
65          27
26          66
DEGREES OF FREEDOM = 1   CHI SQUARE = 33.06912442
42
12
DEGREES OF FREEDOM = 1   CHI SQUARE = 37.12121212
23
14
DEGREES OF FREEDOM = 1   CHI SQUARE = 3.842250414
42
23
DEGREES OF FREEDOM = 1   CHI SQUARE = 11.47058901
12
14
DEGREES OF FREEDOM = 1   CHI SQUARE = 0.735366516
11
14
DEGREES OF FREEDOM = 1   CHI SQUARE = 7.551371636